NASA-CR-194167

THE OHIO STATE UNIVERSITY

Investigation for Improving Global Positioning System (GPS) Orbits Using a Discrete Sequential Estimator and Stochastic Models Of Selected Physical Processes

Clyde C. Goad and C. David Chadwell
Department of Geodetic Science and Surveying

Goddard Space Flight Center
Greenbelt, Maryland 20771

Grant No. NAG 5-2041
Annual Report

August 1993

THE OHIO STATE UNIVERSITY

Investigation for Improving Global Positioning System (GPS) Orbits Using a Discrete Sequential Estimator and Stochastic Models Of Selected Physical Processes

Clyde C. Goad and C. David Chadwell
Department of Geodetic Science and Surveying

Goddard Space Flight Center
Greenbelt, Maryland 20771

Grant No. NAG 5-2041
Annual Report
RF Project No. 760126/726487

August 1993

Introduction

GEODYNII is a conventional batch least-squares differential corrector computer program with deterministic models of the physical environment. Conventional algorithms have been used to process differenced phase and pseudorange data to determine eight-day GPS orbits with several meter accuracy (Schenewerk, 1990). However, random physical processes drive the errors whose magnitudes prevent improving the GPS orbit accuracy. To improve the orbit accuracy, these random processes should be modeled stochastically. The conventional batch least-squares algorithm cannot accommodate stochastic models, only a stochastic estimation algorithm is suitable, such as a sequential filter/smoother. Also, GEODYNII cannot currently model the correlation among data values. Differenced pseudorange, and especially differenced phase, are precise data types that can be used to improve the GPS orbit precision (Counselman et al., 1989). To overcome these limitations and improve the accuracy of GPS orbits computed using GEODYNII, we proposed to develop a sequential stochastic filter/smoother processor by using GEODYNII as a type of trajectory preprocessor. Our proposed processor is now completed. It contains a correlated double difference range processing capability, first order Gauss Markov models for the solar radiation pressure scale coefficient and y-bias acceleration, and a random walk model for the tropospheric refraction correction.

The development approach has been to interface the standard GEODYNII output files (measurement partials and variationals) with software modules containing the stochastic estimator, the stochastic models, and a double differenced phase range processing routine. Thus, no modifications to the original GEODYNII software have been required. A schematic of the development is shown in Figure 1. The observational data are edited in the preprocessor and the data are passed to GEODYNII as one of its standard data types. A reference orbit is determined using GEODYNII as a batch least-squares processor and the GEODYNII measurement partial (FTN90) and variational (FTN80, V-matrix) files are generated. These two files along with a control statement file and a satellite identification and mass file are passed to the filter/smoother to estimate time-varying parameter states at each epoch, improved satellite initial elements, and improved estimates of constant parameters.

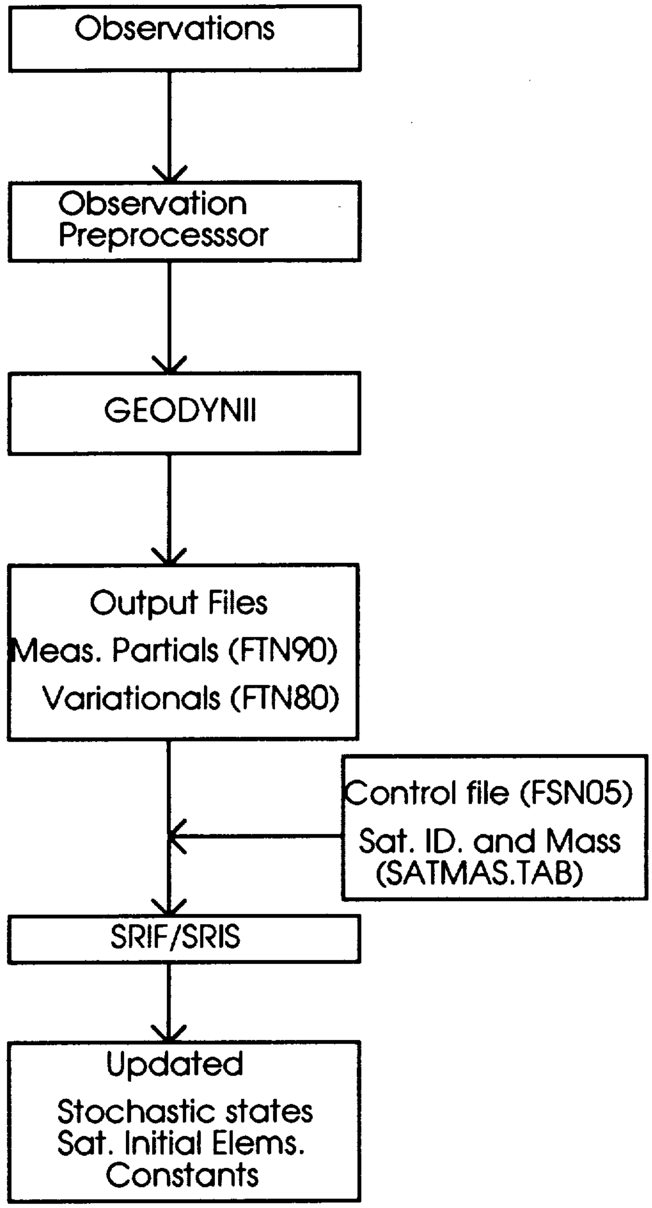

Figure 1. Flowchart showing the procedure as currently developed

Background

The following discussion assumes some familiarity with filtering and smoothing theory as developed, for example in Brown (1983) and Gelb (1974). Additional familiarity is assumed with the square root information filtering and smoothing algorithm as developed in Bierman (1977) and implemented by Swift (1987). Any departures in our implementation from Swift's formulation are

explained here in detail. Some topics are expanded here to clarify and to supplement the discussion in these earlier references. Particular attention has been focused on showing the common foundation of the information and covariance filters. The efficiency of the Householder Transformation to compute an equivalent square upper triangular matrix from a larger rectangular matrix is discussed. Also, the quantities that define the first-order Gauss Markov and random walk models are clearly derived.

The discrete form of the stochastic state equations are:

$$\Delta\mathbf{x}_{j+1} = \Phi_j \Delta\mathbf{x}_j + G\omega_j \tag{1}$$

where

$\Delta\mathbf{x}_j$ is the state at time t_j

$\Phi_j = \Phi(t_{j+1}, t_j)$ is the nonsingular state transition matrix relating the state at t_j to the state t_{j+1}.

ω_j is the vector of white noise process terms with a nonsingular covariance matrix Q_j with $\dim\omega \leq \dim\Delta\mathbf{x}$.

G maps the source white noise process into the state with $\dim\Delta\mathbf{x}$.

The discrete form of the linear measurement model is:

$$\mathbf{z}_j = A_j \Delta\mathbf{x}_j + \mathbf{v}_j \tag{2}$$

where

$\mathbf{z}_j$ is the vector of measurements at time t_j

A_j is the matrix of partial derivatives of the measurement model w.r.t. the state at t_j

$\mathbf{v}_j$ is the vector of measurement noise with the covariance P_o.

The observations are decorrelated and whitened so that $P_o = I$. This is done without a loss of generality. A set of observations with $P_o = I$ can be constructed. The procedure will be given later.

A solution to this problem was first proposed by Kalman in early 1960's (Kalman, 1960; Kalman and Bucy, 1961). A solution for the state and its covariance can be derived by applying Bayes'

rule. This derivation can be found in Maybeck (1979). The results are repeated here with a slight change in notation.

$$\hat{P}_{j+1} = \left[\tilde{P}_j^{-1} + A_j^T A_j\right]^{-1} \tag{3}$$

$$\Delta\hat{x}_j = \hat{P}_{j+1}\left[\tilde{P}_j^{-1}\Delta\tilde{x}_j + A_j^T z_j\right] \tag{4}$$

The '~' symbol refers to the propagated (predicted) estimate of the state and covariance at t_j. The '^' symbol refers to the estimate of the state or covariance after incorporating the measurement at t_j. The state $\Delta\hat{x}_j$ is propagated from t_j. to time t_{j+1}, using equation (1). The covariance $\hat{P}_j$ is propagated to t_{j+1}, by

$$\tilde{P}_{j+1} = \Phi_j \hat{P}_j \Phi_j^T + G_j Q_j G_j^T \tag{5}$$

Notice that the inverse of $\tilde{P}_{j+1}$,, is required in equations (3) and (4). To avoid the inversion of $\tilde{P}_{j+1}$ at each step, a direct propagation of $\tilde{P}_{j+1}^{-1}$ is desired. This can be developed by applying the following lemma to equation (5).

$$\left(A + X^T Y\right)^{-1} = A^{-1} - A^{-1}X^T\left(I + YA^{-1}X^T\right)^{-1} YA^{-1} \tag{6}$$

where

$$A = \Phi_j \hat{P}_j \Phi_j^T$$
$$X^T = G_j Q_j$$
$$Y = G_j^T$$

and defining

$$M_{j+1} = \Phi^T\left(t_{j+1}, t_j\right)\hat{P}_j^{-1}\Phi(t_{j+1}, t_j) \tag{7}$$

$$\tilde{P}^{-1}_{j+1} = M_{j+1} - C_j G_j^T M_{j+1}, \tag{8}$$

where the gain C_j, is :

$$C_j = M_{j+1} G_j \left[G_j^T M_{j+1} G_j + Q_j^{-1} \right]^{-1} \tag{9}$$

To propagate and update the state equations in terms of $\tilde{P}_j^{-1}$ and $\hat{P}_j^{-1}$, the state estimates are replaced by

$$\tilde{y}_j = \tilde{P}_j^{-1} \Delta \tilde{x}_j \tag{10}$$

$$\hat{y}_j = \hat{P}_j^{-1} \Delta \hat{x}_j \tag{11}$$

The state update and propagation equations are:

$$\hat{y}_j = \tilde{y}_j + A_j^T z_j \tag{12}$$

$$\tilde{y}_{j+1} = \left[I - C_j G_j^T \right] \Phi(t_{j+1}, t_j) \hat{y}_j \tag{13}$$

The state estimates can be found at any time by solving equations (10) and/or (11) for $\Delta \tilde{x}_j$ and/or $\Delta \hat{x}_j$. Equations (8) to (13) are an algorithm to solve the problem defined by equations (1) and (2). Here, the inverse covariance is propagated. This algorithm is sometimes called a Bayes' filter. The inverse covariance is also called an information matrix leading to the name information filter. This algorithm requires computing the inverse of an n x n matrix where n is the number of states. The state estimate covariance can be completely uncertain since in the inverse of P_o the elements of the matrix become zero. This algorithm is most efficient when the number of measurements, m, is relatively larger than the number of states, n, and when the solutions for the state and covariance are needed infrequently.

The usual Kalman filter can be derived from equations (3) and (4) by applying the matrix lemma:

$$\left[P^{-1} + A^T A \right]^{-1} = P - PA^T \left[APA^T \right]^{-1} AP \tag{14}$$

These optimal filters, either the Bayes' or Kalman, exhibit numerical instabilities that cause the state estimates to diverge (Bierman and Thornton, 1977). More numerically stable gain matrix expressions have been derived for both the covariance and information matrix forms (Maybeck ibid.). However, these require a significant number of additional matrix computations and are thus not completely satisfactory. A more comprehensive approach was to reformulate the filter algorithm in terms of square roots of the covariance or the information matrix. The square root filter maintains numerical accuracy to approximately the same number of digits with half the word length required by a conventional non-square root algorithm.

The square root (or more correctly the Cholesky factorization) of an *nxn* matrix N is defined as :

$$N = SS^T \tag{15}$$

The square roots are not unique. Any orthogonal transformation (T) of a square root matrix S is also a square root of N. The useful properties of the orthogonal transformation can be shown by factoring $\tilde{P}_j^{1}$ into the product of its Cholesky factors

$$\tilde{P}_j^{1} = \tilde{R}_j^T \tilde{R}_j, \tag{16}$$

and thus, $\hat{P}_j^{1}$ becomes:

$$\hat{P}_j^{1} = \overline{R}_j^T \overline{R}_j = \tilde{R}_j^T \tilde{R}_j + A_j^T A_j \tag{17}$$

where

$$\overline{R}_j = \begin{bmatrix} \tilde{R}_j^T \\ A_j \end{bmatrix} \tag{18}$$

Now

$$\begin{bmatrix} \hat{R}_j \\ 0 \end{bmatrix} = T \begin{bmatrix} \tilde{R}_j^T \\ A_j \end{bmatrix} \tag{19}$$

The transformation T is an orthogonal transformation. Its columns form an orthonormal basis for $\overline{R}_j$ of *n* vectors since $\overline{R}_j$ has rank of *n*. The first *n* vectors span the range space of $\overline{R}_j$ and the vectors *n+1* to *n+m* are orthogonal to this spanning set. Thus, the last *m* rows of $\overline{R}_j$ are zero. Also, the basis vectors were chosen in a manner that $\hat{R}_j$ is upper triangular. A Householder

Transformation T is used to compute $\hat{R}_j$ (Bierman, ibid.). This allows the square root matrices of dimension $(n+m)$ by n to be transformed to an equivalent form of an upper triangular square matrix of dimension n.

Square Root Information Filter and Smoother (SRIF/SRIS)

In application of the Householder Transformation to an augmented matrix is the essence of the Square Root Information Filter (SRIF). The equations are most easily constructed using Bierman's 'data equation' point of view. The problem is treated as a least-squares problem where the least-squares functional is to minimized. This is accomplished by applying a Householder Transformation assuming the state at t_j to be a priori information and augmented with the measurements at t_j. Thus,

$$\hat{T}_j \begin{bmatrix} \tilde{R}_j & \tilde{z} \\ A_j & z_j \end{bmatrix} = \begin{bmatrix} \hat{R}_j & \hat{z}_j \\ 0 & e_j \end{bmatrix} \tag{20}$$

where the 'data equations' are defined as

$$\tilde{z}_j = \tilde{R}_j \Delta\tilde{x}_j \tag{21}$$

$$\hat{z}_j = \hat{R}_j \Delta\hat{x}_j \tag{22}$$

Swift has shown the equivalence of equation (20) with the more conventional formulation of equations (3) and (4).

The propagation of the state and the covariance were given in equations (1) and (5). These can also be incorporated into the SRIF by defining 'data equations' and applying the Householder Transformations. The details of the derivations can be found in Bierman (ibid.). The results are repeated here.

$$\tilde{T}_{j+1} \begin{bmatrix} R_\omega(j) & 0 & \tilde{z}_\omega(j) \\ -\hat{R}_j \Phi_j^{-1} G & \hat{R}_j \Phi_j^{-1} & \tilde{z}_{j+1} \end{bmatrix} = \begin{bmatrix} \tilde{R}_\omega(j+1) & \tilde{R}_{\omega x}(j+1) & \tilde{z}_\omega(j+1) \\ 0 & \tilde{R}_{j+1} & \tilde{z}_{j+1} \end{bmatrix} \tag{23}$$

where the 'data equation' for the noise term ω is

$$z_{\omega}(j) = R_{\omega}(j)\omega(j) \tag{24}$$

Swift has also shown the equivalence of equation (23) to equations (1) and (5).

Bierman further partitions the propagation equation (1) into stochastic states, dynamic states and bias states. Equation (1) can be written as:

$$\begin{bmatrix} \Delta p \\ \Delta x \\ \Delta y \end{bmatrix}_{j+1} = \begin{bmatrix} M & 0 & 0 \\ V_p' & V_x' & V_y' \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta x \\ \Delta y \end{bmatrix}_{j} + \begin{bmatrix} \omega_j \\ 0 \\ 0 \end{bmatrix} \tag{25}$$

where

Δp is the correlated process noise states

Δx states that vary with time by not explicitly influenced by process noise.

Δy bias (constant) parameters.

V_p', V_x', V_y', are transition matrix elements.

The dynamic parameters can be redefined in the form of pseudo epoch state parameters. This dynamic model definition allows the variationals and measurement partials from a batch differential corrector orbit determination program (e.g., GEODYNII) to be used directly in the filter algorithms. The state equations now become

$$\begin{bmatrix} \Delta p \\ \Delta x \\ \Delta y \end{bmatrix}_{j+1} = \begin{bmatrix} M & 0 & 0 \\ V_p & I & 0 \\ 0 & 0 & I \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta x \\ \Delta y \end{bmatrix}_{j} + \begin{bmatrix} \omega_j \\ 0 \\ 0 \end{bmatrix} \tag{26}$$

where

$V_p = V_p(t_{j+1}, t_j) = V_x'^{-1}(t_{j+1}, T_0) V_p'(t_{j+1}, t_j)$

$V_x'^{-1}(t_{j+1}, T_0)$ is the inverse of the state transition matrix interpolated from the GEODYNII V-matrix file (FTN80)

$V_p'(t_{j+1}, t_j)$ is the transition matrix of the time-varying parameters from t_j to t_{j+1}.

The transformation in equation (20) is written as a two step transformation that saves storing a block of *ny* x *(np+nx)* zeroes that would be present if equation (20) was used in its original form.

$$\mathrm{T}_{px}\begin{bmatrix} \tilde{\mathrm{R}}_p & \tilde{\mathrm{R}}_{px} & \tilde{\mathrm{R}}_{py} & \tilde{\mathrm{z}}_p \\ \tilde{\mathrm{R}}_{xp} & \tilde{\mathrm{R}}_x & \tilde{\mathrm{R}}_{xy} & \tilde{\mathrm{z}}_x \\ \mathrm{A}_p & \mathrm{A}_x & \mathrm{A}_y & \mathrm{z} \end{bmatrix} = \begin{bmatrix} \hat{\mathrm{R}}_p & \hat{\mathrm{R}}_{px} & \hat{\mathrm{R}}_{py} & \hat{\mathrm{z}}_p \\ 0 & \hat{\mathrm{R}}_x & \hat{\mathrm{R}}_{xy} & \hat{\mathrm{z}}_x \\ 0 & 0 & \hat{\mathrm{A}}_y & \hat{\mathrm{z}} \end{bmatrix} \tag{27}$$

$$\hat{\mathrm{T}}_y\begin{bmatrix} \tilde{\mathrm{R}}_y & \tilde{\mathrm{z}}_y \\ \hat{\mathrm{A}}_y & \hat{\mathrm{z}} \end{bmatrix} = \begin{bmatrix} \hat{\mathrm{R}}_y & \hat{\mathrm{z}}_y \\ 0 & \mathrm{e} \end{bmatrix} \tag{28}$$

The mapping equation (23) becomes, assuming z_ω is zero

$$\tilde{\mathrm{T}}_p\begin{bmatrix} -\mathrm{R}_\omega \mathrm{M} & \mathrm{R}_\omega & 0 & 0 & 0 \\ \hat{\mathrm{R}}_p - \hat{\mathrm{R}}_{px}\mathrm{V}_p & 0 & \hat{\mathrm{R}}_{px} & \hat{\mathrm{R}}_{py} & \hat{\mathrm{z}}_p \\ -\hat{\mathrm{R}}_x\mathrm{V}_p & 0 & \hat{\mathrm{R}}_x & \hat{\mathrm{R}}_{xy} & \hat{\mathrm{z}}_x \end{bmatrix}_j = \begin{bmatrix} \mathrm{R}^*_p & \mathrm{R}^*_{pp} & \mathrm{R}^*_{px} & \mathrm{R}^*_{py} & \mathrm{z}^*_p \\ 0 & \tilde{\mathrm{R}}_p & \tilde{\mathrm{R}}_{px} & \tilde{\mathrm{R}}_{py} & \tilde{\mathrm{z}}_p \\ 0 & \tilde{\mathrm{R}}_{xp} & \tilde{\mathrm{R}}_x & \tilde{\mathrm{R}}_{xy} & \tilde{\mathrm{z}}_x \end{bmatrix}_{j+1} \tag{29}$$

The subroutine in Bierman (ibid., 155-157) neglected the upper triangular elements of R^*_p above the diagonal. A subroutine to compute right side of equation (29) including the neglected off-diagonal elements of R^*_p is given in Appendix B.

The smoothing process is a backward filter of the forward filter results. For the orbit determination problem, the fixed interval smoother is appropriate. For Kalman filtering, the Rauch-Tung-Streibel (RTS) smoothing algorithm is widely used (Brown, ibid.). A general formulation for inverse covariance smoothing is given by Maybeck (ibid.). The Square Root Information Smoother (SRIS) is given by Bierman (ibid.). The equation for the implementation of Bierman's pseudo epoch formulation is given by Swift (ibid.). Swift also shows the equivalence of the SRIS to the RTS smoother. The SRIS equation is

$$\mathrm{T}^*_{ppx}\begin{bmatrix} \mathrm{R}^*_{pp} & \mathrm{R}^*_p + \mathrm{R}^*_{pp}\mathrm{M} + \mathrm{R}^*_{px}\mathrm{V}_p & \mathrm{R}^*_{px} & \mathrm{R}^*_{py} & \mathrm{z}^*_p \\ \mathrm{R}^\#_p & \mathrm{R}^\#_p\mathrm{M} + \mathrm{R}^\#_{px}\mathrm{V}_p & \mathrm{R}^\#_{px} & \mathrm{R}^\#_{py} & \mathrm{z}^\#_p \\ 0 & \mathrm{R}^\#_x\mathrm{V}_p & \mathrm{R}^\#_x & \mathrm{R}^\#_{xy} & \mathrm{z}^\#_x \end{bmatrix}_{j+1} = \begin{bmatrix} \mathrm{R}'_{pp} & \mathrm{R}'_p & \mathrm{R}'_{px} & \mathrm{R}'_{py} & \mathrm{z}'_p \\ 0 & \mathrm{R}^\#_p & \mathrm{R}^\#_{px} & \mathrm{R}^\#_{py} & \mathrm{z}^\#_p \\ 0 & 0 & \mathrm{R}^\#_x & \mathrm{R}^\#_{xy} & \mathrm{z}^\#_x \end{bmatrix}_j \tag{30}$$

The top row of the matrix on the right side is not needed again, but the other terms are combined with the smoothing coefficients at t_j to smooth back to t_{j-1}.

From the general expression of the data equation, and the relationship of the covariance to the square roots of the information matrix, the solution for the states and their covariances for either the filter or smoothing operations are

$$\mathbf{x}_j = \mathbf{R}_j^{-1}\mathbf{z}_j \tag{31}$$

$$\mathbf{P}_j = \mathbf{R}_j^{-1}\mathbf{R}_j^{-\mathrm{T}} \tag{32}$$

For filtering, the right sides of equations (27) and (28) are solved for the unknowns and the covariances as

$$\begin{bmatrix} \Delta \mathrm{p} \\ \Delta \mathrm{x} \\ \Delta \mathrm{y} \end{bmatrix}_j = \begin{bmatrix} \begin{bmatrix} \mathrm{R}_x & \mathrm{R}_{px} \\ 0 & \mathrm{R}_x \end{bmatrix}^{-1} \left[\begin{pmatrix} \mathrm{z}_p \\ \mathrm{z}_x \end{pmatrix} - \begin{pmatrix} \mathrm{R}_y \\ \mathrm{R}_{xy} \end{pmatrix} \mathrm{R}_y^{-1}\mathrm{z}_y \right] \\ \mathrm{R}_y^{-1}\mathrm{z}_y \end{bmatrix} \tag{33}$$

$$\mathrm{R}_j^{-1} = \begin{bmatrix} \begin{bmatrix} \mathrm{R}_p & \mathrm{R}_{px} \\ 0 & \mathrm{R}_x \end{bmatrix}^{-1} & -\begin{bmatrix} \mathrm{R}_p & \mathrm{R}_{px} \\ 0 & \mathrm{R}_x \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{R}_{px} \\ \mathrm{R}_{xy} \end{bmatrix} \mathrm{R}_y^{-1} \\ 0 & \mathrm{R}_y^{-1} \end{bmatrix}_j \tag{34}$$

$$\mathrm{P}_{px} = \begin{bmatrix} \mathrm{R}_p & \mathrm{R}_{px} \\ 0 & \mathrm{R} \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{R}_p & \mathrm{R}_{px} \\ 0 & \mathrm{R}_x \end{bmatrix}^{-\mathrm{T}} + \begin{bmatrix} \mathrm{R}_p & \mathrm{R}_{px} \\ 0 & \mathrm{R}_x \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{R}_{py} \\ \mathrm{R}_{xy} \end{bmatrix} \mathrm{R}_y^{-1} \left[\begin{bmatrix} \mathrm{R}_p & \mathrm{R}_{px} \\ 0 & \mathrm{R}_x \end{bmatrix}^{-1} \begin{bmatrix} \mathrm{R}_{py} \\ \mathrm{R}_{xy} \end{bmatrix} \mathrm{R}_y^{-1} \right]^{\mathrm{T}} \tag{35}$$

$$\mathrm{P}_y = \mathrm{R}_y^{-1}\mathrm{R}_y^{-\mathrm{T}} \tag{36}$$

For the smoothing problem, the right side of (30) is used in equations (32)-(36) with the smoothed value of $\mathbf{z}_y$ and $\mathbf{R}_y$. The smoothed values of $\mathbf{z}_y$ and $\mathbf{R}_y$ are the values at the last filter step.

Time-Varying Stochastic Parameter Models

Many physical processes can be modeled using one of the Gauss Markov filters easily constructed by filtering white noise through a simple filter. The processes that are of interest here are the

first-order Gauss Markov model and the random walk model. The discrete mathematical expression for these models is now derived from their continuous forms. The first-order Gauss Markov process describes the physical process where the state at t_{j+1} depends only on the previous state at t_j. This process can be described by the stochastic differential equation.

$$\frac{dp(t)}{dt} - \frac{1}{\tau} p(t) = \omega \tag{37}$$

where

τ is the correlation time

ω is the white Gaussian noise with zero mean and covariance Q,

$E(\omega_j) = 0, E(\omega_j, \omega_k) = Q\delta(j-k) = q_{con}\delta(j-k)$ where q_{con} is the continuous spectral density.

The variance of p, σ_p^2 satisfies the differential equation

$$\frac{d\sigma_p^2(t)}{dt} = -\frac{2}{\tau}\sigma_p^2(t) + q_{con} \tag{38}$$

If the process is allowed to continue for a time interval several multiples longer than τ, the $E(p^2(t))$ will approach a limit and $\frac{d\sigma_p^2(t)}{dt}$ will approach zero. This is the steady state variance. By setting $\frac{d\sigma_p^2(t)}{dt} = 0$ and solving equation (38) for steady state $\sigma_p^2(t_{st})$ is found

$$\sigma_p^2(t_{st}) = \frac{\tau}{2} q_{con} \tag{39}$$

From state space methods the solution of equation (37) is

$$p(j+1) = M(j+1, j)p(j) + \int_j^{j+1} M(j, \lambda)\omega(\lambda)d\lambda \tag{40}$$

and covariance of p is

$$C_p = MP_jM^T + \int_j^{j+1} M(j+1, \lambda)Q(\lambda)M^T(j+1, \lambda)d\lambda \tag{41}$$

The matrix M is the state transition matrix which must satisfy the relationships

$$\dot{M} = -\frac{1}{\tau}M \qquad \text{and} \qquad M(j,j) = \mathrm{I} \tag{42}$$

The solution for M is

$$M = e^{-\frac{\Delta t}{\tau}} \tag{43}$$

where $\Delta t = t_{j+1} - t_j$

Thus the discrete form of the state update becomes

$$\Delta p(j+1) = M(j+1,j)\Delta p(j) + \omega_j \tag{44}$$

This is the top row of equation (26).

Now the solution for the discrete covariance update is derived from

$$q_{dis} = \int_{t_t}^{t_{j+1}} q_{con} e^{-\frac{2(\lambda - t_j)}{\tau}} d\lambda = -q_{con}\frac{\tau}{2} e^{-\frac{2(\lambda - t_j)}{\tau}} \Bigg|_{t_j}^{t_{j+1}} = q_{con}\frac{\tau}{2}\left(1 - e^{-\frac{2\Delta t}{\tau}}\right) \tag{45}$$

The random walk model is a special case of the first order Gauss Markov where $\tau \to \infty$. The discrete state update becomes

$$\Delta p(j+1) = \Delta p(j) + \omega_j \tag{46}$$

The discrete covariance is found from

$$q_{dis} = \lim_{\tau \to \infty} q_{con}\frac{\tau}{2}\left(1 - e^{-\frac{2\Delta t}{\tau}}\right) = q_{con}\lim_{\tau \to \infty}\frac{\tau}{2}\left[1 - \left(1 - \frac{2\Delta t}{\tau} + \frac{\left(\frac{2\Delta t}{\tau}\right)^2}{2!} - \ldots\right)\right]$$

$$= q_{con} \lim_{\tau \to \infty} \left(\Delta t + \sum_{i=1}^{\infty} \text{constant}_i * \left(\frac{1}{\tau} \right)^i \right) = q_{con} \Delta t \tag{47}$$

So, to implement the first-order Gauss Markov process, the correlation time (τ) and the continuous process noise variance (q_{con}) must be specified. The matrix M is computed using equation (43) and the q_{dis} from equation (45). The random walk model is specified by defining the continuous process noise variance (q_{con}), here $M = \mathrm{I}$, and q_{dis} is computed using equation (47).

Solar Radiation Pressure Scale Coefficient and Y-bias Acceleration Model

The orbit related stochastic parameters modeled in OSUORBFS are first-order Gauss Markov models for the solar radiation pressure scale coefficient and the y-bias acceleration. The V_p matrix that maps the effects of the stochastic parameters $\Delta \mathrm{p}_j$, on the epoch state parameters $\Delta \mathbf{x}_j$ is derived following the scheme of Swift (ibid.). The V_p matrix has dimension nx (number of pseudo epoch state parameters) by nd (number of orbit-related stochastic parameters) and has the general form of

$$\mathrm{V}_p(t_{j+1}, t_j) = \begin{bmatrix} \Phi^1_{K_r} & 0 & \cdots & \cdots & 0 \\ 0 & \ddots & \ddots & & \vdots \\ \vdots & \ddots & \Phi^i_{K_r} & \ddots & \vdots \\ \vdots & & \ddots & \ddots & 0 \\ 0 & \cdots & \cdots & 0 & \Phi^{nsat}_{K_r} \end{bmatrix} \tag{48}$$

The ith satellite contribution $\Phi^i_{K_r} = \Phi^i_{K_r}(t_{j+1}, t_j)$, a 6 by 2 matrix, to V_p is computed as

$$\Phi^i_{\mathrm{K}_r}(t_{j+1}, t_j) = \Phi_e^{-1}(t_{j+1}, t_j) \begin{bmatrix} \dfrac{\partial \mathrm{r}(t_{j+1})}{\partial \mathrm{K}_{r_1}(t_j)} & \dfrac{\partial \mathrm{r}(t_{j+1})}{\partial \mathrm{K}_{r_2}(t_j)} \\ \dfrac{\partial \dot{\mathrm{r}}(t_{j+1})}{\partial \mathrm{K}_{r_1}(t_j)} & \dfrac{\partial \dot{\mathrm{r}}(t_{j+1})}{\partial \mathrm{K}_{r_2}(t_j)} \end{bmatrix} \tag{49}$$

where

$\Phi_e^{-1}(t_{j+1},t_j)$ is the state transition matrix interpolated from the GEODYNII output V-matrix file (FTN80) variationals. See also equation (26).

The partial derivatives of the position and velocity at t_{j+1} with respect to the solar radiation pressure scale coefficient and the y-bias acceleration at t_j are approximated using a second-order Taylor Series expansion.

$$\frac{\partial \mathbf{r}(t_{j+1})}{\partial \mathrm{K}_{r_l}(t_j)} = \frac{\Delta t^2}{2}\frac{\partial \ddot{r}(t_j)}{\partial \mathrm{K}_{r_l}(t_j)}$$

$$\frac{\partial \dot{\mathbf{r}}(t_{j+1})}{\partial \mathrm{K}_{r_l}(t_j)} = \Delta t\frac{\partial \ddot{r}(t_j)}{\partial \mathrm{K}_{r_l}(t_j)} - \frac{\Delta t^2}{2}\frac{\partial \ddot{r}(t_j)}{\partial \mathrm{K}_{r_l}(t_j)}\mathrm{B}_{\mathrm{K}_{r_l}} \quad l = 1,2 \tag{50}$$

where

$$\mathrm{B}_{\mathrm{K}_r} = \left[\mathrm{B}_{\mathrm{K}_{r_1}} \;\vdots\; \mathrm{B}_{\mathrm{K}_{r_2}}\right] = \begin{bmatrix} -\frac{1}{\tau_{\mathrm{K}_{r_1}}} & 0 \\ 0 & -\frac{1}{\tau_{\mathrm{K}_{r_2}}} \end{bmatrix}$$

$\mathrm{B}_{\mathrm{K}_r} = 0$ for a random walk.

Since the angle between the x and y satellite axes is not easily estimated it is not modeled in OSUORBFS. A 90 degree angle is assumed. Thus the following differs slightly from Swift.

$$\left[\frac{\partial \ddot{r}(t_j)}{\partial \mathrm{K}_{r_1}(t_j)} \;\vdots\; \frac{\partial \ddot{r}(t_j)}{\partial \mathrm{K}_{r_2}(t_j)}\right] = \mathrm{R}_s \begin{bmatrix} a_x^m shape & 0 \\ 0 & shape \\ a_z^m shape & 0 \end{bmatrix} \tag{51}$$

where

a_x^m and a_z^m are the accelerations along the satellite x and z axis respectively.

The ROCK4 and ROCK42 models are used to compute a_x^m and a_z^m.

shape is either 0 or 1 depending if the sun is obstructed by the earth from view of

the satellite.

R_s is a matrix transformation from the satellite axis system to the True of Reference Date (TORD) inertial Cartesian reference system.

The ROCK4 and ROCK42 models (Fliegel and Gallini, 1992), the matrix transformation R_s, and the computation of *shape* require the sun-earth-satellite positions. The mean of date (MOD) positions of the sun and earth were computed using the closed form expressions of Fliegel and Harrington (1993).

Tropospheric Refraction Correction

The measurement-related stochastic parameter modeled in OSUORBFS is a random walk model for the refraction correction (Tralli et. al., 1988; Herring et. al., 1990). This model is defined by equations (46) and (47).

Double Difference Observable Decorrelation and Whitening

The full covariance matrix for the double difference range data is constructed. The observations are then decorrelated and whitened. The general form of the observation equations as defined in equation (2) is

$$z = Ax + v \tag{52}$$

Here, the observation error v has a zero mean, $E(v) = 0$, but is correlated, $E(vv^T) = P_v$

A set of uncorrelated observations with unit covariance can be constructed from the lower triangular square root of P_v.

$$P_v = L_v L_v^T \tag{53}$$

Here, L_v can be computed by a lower Cholesky factorization of P_v. The desired independent set of observations is

$$L_v^{-1} z = L_v^{-1} Ax + L_v^{-1} v \tag{54}$$

At any particular epoch, the m = *(#stations-1)*x*(#satellites-1)* linear independent double difference range data types can be formed. These m observations are independent in the sense of linear algebra, but are statistically correlated. Each of the m observations has the form (GEODYNII measurement type=87)

$$[(s1 \to t1) - (s2 \to t1)] - [(s1 \to t2) - (s2 \to t2)] \tag{55}$$

where

$(s1 \to t1)$ etc., are the satellite-station range observations (e.g., satellite 1 to station 1).

For small regional networks, a single satellite station pair is selected as the base satellite-station and the m observations are constructed by differencing the remaining satellite-stations with the base pair. For a global network, the distance between stations may prevent using a single base pair to construct all observations at that epoch. Thus, no consistent numerical structure exists that would permit a symbolic construction of the decorrelated measurement set. P_v and L_v^{-1} must be computed numerically at each epoch. P_v is computed using conventional error propagation

$$P_v = \sigma_r^2 G G^T \tag{56}$$

where

σ_r^2 is the standard deviation of the single one-way range measurement

G is the matrix of partial derivatives of the observation equation with respect to the one-way range. This matrix contains elements of -1,0,1 which are the linear combination of one-way ranges that define the double difference

The decorrelated observation set with unit covariance is obtain from equations (53) and (54).

OSUORBFS

The program OSUORBFS is designed to filter and smooth the GEODYNII batch solutions. OSUORBFS requires from GEODYNII the measurement partials file (FTN90) and the variationals V-matrix file (FTN80). The user must supply a user input file (FSN05) and a file of satellite identification numbers and masses (SATMAS.TAB). The GEODYNII processing proceeds in the usual way with TDF, G2S, and G2E program executions. On the last iteration an

output of the setup deck (FTN05) which contains the current parameter estimates is requested using PUNCH to output the new setup deck in file FTN07. Then FTN07 is modified to include global cards PARFIL and EMATRX to output files FTN90 and FTN80. The maximum iteration numbers for the global (outer) and arc (inner) are set to one on the ENDGLB and REFSYS statements.

Now TDF, G2S, and G2E are executed and FTN06, FTN80, and FTN90 files are output. An alternative approach avoiding the restart of GEODYNII is to force an additional iteration in the first GEODYN execution by increasing the outer/inner iteration maximum counts and decreasing the RMS tolerance. This approach would be less cumbersome to implemented. Additional operational experience is needed to determine which approach is satisfactory.

The user must construct the control file, FSN05, for OSUORBFS. This file contains six control statements (REFSYS, DECORR, FILSMT, UPDTRJ, CONPRT, SATMAS) to control the configuration of the filter/smoother solutions. The six statements are mandatory. These statements are explained in Appendix A.

FSN05 must also contain the parameter labels from FTN06. The measurement partials and variationals form GEODYNII are identified by internal parameter labels as described in the GEODYNII manual volume 5. For OSUORBFS to recognize these partials, the parameter labels as they appear in FTN80 and FTN90 must be specified in FSN05. These labels can be accessed by printing the EMATRIX header record in FTN06 during the last iteration of the GEODYN run. They must be manually edited and placed in FSN05.

FSN05 must also contain the parameter types as defined in the following table.

Type #	Description	Example
1	orbit-related stochastic	solar radiation pressure (1st Gauss Markov) y-bias acceleration (1st Gauss Markov)
2	measurement-related stochastic	tropospheric refraction correction (random walk)
3	pseudoepoch state	satellite initial elements
4	measurement-related constant	double difference bias, tropospheric refraction correction
5	orbit-related constant	solar radiation pressure coefficient, y-bias

For each parameter type the apriori standard deviation must be specified. Additionally, for the first-order Gauss Markov model, the continuous process noise standard deviation ($\sqrt{q_{con}}$) and the correlation (τ) time must be specified. For the random walk model, the continuous process noise standard deviation ($\sqrt{q_{con}}$) and a negative correlation time ($-\tau$, which acts as a flag) must be specified. These are read in a free format.

The order of the parameter types in FSN05 is arbitrary; the file is sorted and the time-varying stochastic parameters are moved to the top of the file to accommodate the space saving implementation of the V_p as an *nx* x *nd* matrix.

The stochastic parameters (types 1, 2) are assumed to be zero mean processes. Typically the physical process modeled is not zero mean. The non-zero mean is estimated as a constant (types 4, 5). Thus, the constant (types 4, 5) and the stochastic parameters (types 1, 2) are estimated together. The constant can be estimated without a stochastic parameter, but a stochastic parameter must be estimated with a constant unless of course the process has a zero mean.

Since GEODYNII does not have time-varying models the physical processes are modeled by estimating constants over consecutive segments of time. For example, the tropospheric scale correction in GEODYNII may be modeled over a 24 hour period by estimating a constant over the first 12 hours and another constant over the second 12 hours. In OSUORBFS, the one constant and a time-varying stochastic parameter would be estimated over the entire span of 24 hours. This requires the measurement partials from the two consecutive constant estimates in GEODYNII to be concatenated. This is controlled by the CONPRT statement.

OSUORBFS can be implemented as a conventional least-squares sequential estimator by specifying all parameters as pseudo epoch state (Type 3) and constant parameters (Types 4, 5). The partial file should not be concatenated. The full covariance matrix may be generated. If the GEODYNII solution is to be repeated using OSUORBFS in a sequential least-squares step, then the full covariance matrix should not be formed.

REFERENCES

Bierman, G. J., (1977): *Factorization Methods for Discrete Sequential Estimation*, Academic Press, New York, New York.

Bierman, G. J., C. L. Thornton, (1977): Numerical Comparison of Kalman Filter Algorithms: Orbit Determination Case Study, Automatica, **13**, 23-35.

Brown, R. G., (1983): *Introduction to Random Signal Analysis and Kalman Filtering*, John Wiley and Sons, New York, New York.

Counselman III, C. C., R. I. Abbot, (1989): Method of Resolving Radio Phase Ambiguity in Satellite Orbit Determination, *Journal of Geophysical Research*, **94**, B6, 7058-7064.

Gelb, A., *Ed.*(1974): *Applied Optimal Estimation*, Massachusetts Institute of Technology, Cambridge, Massachusetts.

Herring, T. A., J. L. Davis, I. I. Shapiro, (1990): Geodesy by Radio Interferometry: The Application of Kalman Filtering to the Analysis of Very Long Baseline Interferometry Data, *Journal of Geophysical Research*, **95**, B8, 12561-12581.

Kalman, R. E., (1960): A New Approach to Linear Filtering and Prediction Problems, *Journal of Basic Engineering* (ASME), **82D**, 34-35.

Kalman, R. E., R. Bucy, (1961): New Results in Linear Filtering and Prediction, *Journal of Basic Engineering* (ASME), **83D**, 95-108.

Fliegel, H. F., T. E. Gallini (1992): Global Positioning System Radiation Force Model for Geodetic Applications, *Journal of Geophysical Research*, **97**, B1, 559-568.

Fliegel, H. F., K. M. Harrington (1993): Closed Form Sun, Moon, and Nutation Algorithms: Application for Trace and Global Positioning System, ATR-93(3473)-1, The Aerospace Corporation, El Sequndo, California.

Maybeck, P. S., (1979): *Stochastic Models, Estimation, and Control: Volume 2*, Academic Press, New York, New York.

Schenewerk, M. S., G. L. Mader, M. Chin, W. Kass, R. Dulaney, J. R. MacKay, R. H. Foote, (1990): Status of the CIGNET and Orbit Determination at the National Geodetic Survey, Proceedings of the Second International Symposium on Precise Positioning with the Global Positioning System, Sept. 3-7, 1990, Ottawa, Canada, 179-189.

Swift, E. R., (1987): Mathematical Description of the GPS Multisatellite Filter/Smoother, NSWC TR87-187, Naval Surface Warfare Center, Dahlgren, Virginia.

Tralli, D. M., T. H. Dixon, S. A., Stephens, (1988): Effect of Wet Tropospheric Path Delays on Estimation of Geodetic Baselines in the Gulf of California Using the Global Positioning System, *Journal of Geophysical Research*, **93**, B6, 1988, 6545-6557.

APPENDIX A

REFSYS

```
----+----1----+----2----+----3----+----4----+----5----+----6----
REFSYS              910208000000.0000000
----+----1----+----2----+----3----+----4----+----5----+----6----
```

COLUMNS	FORMAT	DESCRIPTION	UNITS
1-6	A6	REFSYS - Specifies the True of Reference Date (TORD) reference system used by GEODYNII. The time <u>MUST</u> be the same as the time used on the REFSYS statement in FTN05 (G2S). Only TORD is valid. Mean of Date (MOD) J2000 is not currently implemented	
7	blank		
21-26	I6	Year,month,day of reference date (YYMMDD)	
27-30	I4	Hour, minute of reference date (HHMM)	
31-40	D10.8	Seconds of reference date (SS.sssssss)	

DECORR

```
----+----1----+----2----+----3----+----4----+----5----+----6----
DECORR 1     0.10
----+----1----+----2----+----3----+----4----+----5----+----6----
```

COLUMNS	FORMAT	DESCRIPTION	UNITS
1-6	A6	DECORR - controls the computation of the full covariance matrix and decorrelation for double difference ranges	
7	blank		
8	I1	= 0, measurements assumed uncorrelated, measurements whitened by dividing by the GEODYNII supplied weight. = 1, full covariance matrix computed for double differenced ranges, then decorrelated and whitened.	
9-10	blank		
11-20	D10.5	standard deviation for a one-way range measurement	meters

FILSMT

```
----+----1----+----2----+----3----+----4----+----5----+----6----
FILSMT 1111111111
----+----1----+----2----+----3----+----4----+----5----+----6----
```

COLUMNS	FORMAT	DESCRIPTION	UNITS
1-6	A6	FILSMT - controls the filter/smoother operations, controls the computation of the estimates/covariances of the stochastic and pseudo epoch parameters (px) and the constants (y)	
7	blank		
8	I1	= 0, do not filter data = 1, filter data	
9	I1	= 0, do not compute the estimate px at each filter step = 1, compute the estimate px at each filter step	
10	I1	= 0, do not compute the covariance px at each filter step = 1, compute the covariance px at each filter step	
11	I1	= 0, do not compute the estimate y at each filter step = 1, compute the estimate y at each filter step	
12	I1	= 0, do not compute the covariance y at each filter step = 1, compute the covariance y at each filter step	
13	I1	= 0, do not smooth data = 1, smooth data	
14	I1	= 0, do not compute the estimate px at each smoother step = 1, compute the estimate px at each smoother step	

15	I1	= 0, do not compute the covariance px at each smoother step = 1, compute the covariance px at each smoother step
16	I1	= 0, do not compute the estimate y at the last filter step/first smoother step = 1, compute the estimate y at the last filter step/first smoother step
17	I1	= 0, do not compute the covariance y at the last filter step/first smoother step = 1, compute the covariance y at the last filter step/first smoother step

UPDTRJ

```
----+----1----+----2----+----3----+----4----+----5----+----6----
UPDTRJ 1
----+----1----+----2----+----3----+----4----+----5----+----6----
```

COLUMNS	FORMAT	DESCRIPTION	UNITS
1-6	A6	UPDTRJ - controls the satellite trajectory computation and output in a TORD system to file	
7	blank		
8	I1	= 0, trajectory is not updated = 1, trajectory is updated	

CONPRT

```
----+----1----+----2----+----3----+----4----+----5----+----6----
CONPRT 1
----+----1----+----2----+----3----+----4----+----5----+----6----
```

COLUMNS	FORMAT	DESCRIPTION	UNITS
1-6	A6	CONPRT - controls the concatenation of the piece-wise measurement partials to the first partial location	
7	blank		
8	I1	= 0, no concatenation = 1, concatenate	

SATMAS

```
----+----1----+----2----+----3----+----4----+----5----+----6----
SATMAS 0
----+----1----+----2----+----3----+----4----+----5----+----6----
```

COLUMNS	FORMAT	DESCRIPTION	UNITS
1-6	A6	SATMAS - controls which satellite number system to use	
7	blank		
8	I1	= 0, GEODYN international satellite id =1, OSU modified international satellite id	

APPENDIX B

```
      subroutine hsttp(dt,np,nx,npx,nd,ntot,tau,vp,rw,s,rpsm,v,dm,maxnd,
     .                  maxnx,maxnp,maxobs,maxntr,maxntc,pnstdv)
c=======================================================================
c        Apply a sequence of elementary Householder transformations to
c        partially triangularize the a posteriori information array,
c        this process propagates the SRIF from time t to time t+dt.
c
c        Adapted from 'Factorization Methods for Discrete Sequential
c                      Estimation' by Gerald J. Bierman, pp.155-157.
c
c        Version        Comments                         Pgmr.
c         9305.1     Modified to compute correctly the    D. Chadwell
c                    Rp* matrix. The off-diagonal elements
c                    were missing from original version.
c
c
c
c        Variable     Type  i/o    Description
c
c          dt         r*8    i     Propagation interval (secs.)
c          np         i*4    i     Number of stochastic time-varying
c                                    parameters
c          nx         i*4    i     Number of bias (constant) parameters
c          nd         i*4    i     Number of orbit-related stochastic
c                                    paramters
c         tau(np)     r*8    i     Correlation times
c        Vp(nx,nd)    r*8    i     The first Nd columns of the Vp matrix
c                                    correspond to the dynamic paramters;
c                                    the last Np-Nd columns are omitted
c                                    because they are in theory zero.
c         Rw(np)      r*8    i     Process noise standard deviation
c                                    reciprocals
c    S(nx+2*np,ntot) r*8     i     The top Np+Nx rows of S contain the
c                                    SRIF array corresponding to the p
c                                    and x variables; the bottom p rows
c                                    are used to store smoothing related
c                                    terms.
c                            o     Time-updated array with smoothing-
c                                    related terms stored in the bottom
c                                    portion of S
c   Rpsm(np*(np+1)/2)r*8     i     Upper triangular matrix contains
c                                    smoothing related terms from the
c                                    t-dt to t.
c                            o     Upper triangular matrix contains
c                                    smoothing related terms from t to
c                                    t+dt.
c
c                      S on input:                       S on output:
c
c                    Npx
c                ------------
c                 Np    Nx  |  Ny  |  1
c       |        -------------------------  |      -----------------------
c       | Np | ^Rp    ^Rpx | ^Rpy | ^Zp | |       |~Rp   ~Rpx | ~Rpy | ~Zp |
c Tp    | Nx | ^Rpx   ^Rx  | ^Rxy | ^Zx | |  ==>  |~Rxp  ~Rx  | ~Rxy | ~Zx |
c       |        -------------------------  |      -----------------------
c       | Np |   0     0   |  0   |  Zw | |       |*Rpp *Rpx | *Rpy | *Zp |
c       |        -------------------------  |      -----------------------
c
c-----------------------------------------------------------------------
      implicit none
      integer*4                i,j,k,l,np,nd,nx,npx,ntot,j1,j2
      integer*4                maxnd,ict,index,maxnx,idiag,idiag2
      integer*4                maxnp,maxobs,maxntr,maxntc
```

```
      double precision       vp(maxnx,maxnd)
      double precision       s(maxntr,maxntc)
      double precision       v( maxnp+maxnx+maxobs )
      double precision       rw(maxnp),tau(maxnp),dm(maxnp),z,sigma
      double precision       alpha,delta,dt,tmp
      double precision       rpsm( maxnp*(maxnp+1)/2 )
      double precision       pnstdv( maxnp )


      z=0.d0
      do j=1,np
         if(tau(j).gt.0.d0)then     !!Rw for 1st order Gauss Markov Model
            dm(j)=dexp( -dt/tau(j) )
            rw(j)=1.d0/( pnstdv(j)*dsqrt(1.d0-dm(j)*dm(j)) )
         else                       !!Rw for random walk model
            dm(j)=1.d0              !!If tau.lt.0 flag for random walk
            rw(j)=dsqrt( 1.d0/dt )/pnstdv(j)
         endif
      enddo

      idiag=0
      idiag2=0
      do j1=1,np
         idiag2=idiag2+j1
         if(j1.le.nd)then
            do i=1,npx
               do k=1,nx
                  s(i,1)=s(i,1)-s(i,np+k)*vp(k,j1)
               enddo
            enddo
            if(j1.gt.1)then
               ict=npx+j1-1
               idiag=idiag+j1
               index=idiag
               do i=j1,np
                  do k=1,nx
                     rpsm(index)=rpsm(index)-s(ict,np+k)*vp(k,j1)
                  enddo
                  index=index+i
               enddo
            endif
         endif

         alpha = -rw(j1)*dm(j1)  !!  Assumes an uncorrelated process noise
                                 !!  cov.
         sigma = alpha*alpha
         do i=1,npx
            v(i)=s(i,1)
            sigma = sigma + v(i)*v(i)
         enddo
         sigma = dsqrt( sigma )
         alpha = alpha - sigma
         ict=j1-1
         index=idiag2
         rpsm(index) = sigma
         sigma = 1.d0/(sigma*alpha)
         do j2=2,ntot
            delta=z
c           if(j2.eq.ntot)delta=alpha*zw(j1)  !! Assume zero mean
            do i=1,npx
               delta = delta + s(i,j2)*v(i)
            enddo
            delta=delta*sigma
            l=j2-1
```

```
            tmp=delta*alpha
            if(j2.gt.np)then
               l=j2
            else
               if( j2.le.(np+1-j1) )then
                  ict=ict+1
                  index=index+ict
                  rpsm( index ) = tmp
               endif
            endif
            s(npx+j1,l)=tmp

            do i=1,npx
               s(i,l)=s(i,j2)+delta*v(i)
            enddo
         enddo
c        s(npx+j1),ntot)=s(npx+j1,ntot)+delta*zw(j1)   !! Assume zero
                                                        !! mean
         delta=alpha*rw(j1)*sigma
         s(npx+j1,j1)=rw(j1)+delta*alpha
         do i=1,npx
            s(i,np)=delta*v(i)
         enddo
      enddo
      return
      end
```

The Ohio State University
Research Foundation
1960 Kenny Road
Columbus, Ohio 43210-1063
614-292-3805